

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89123610.1

(8) Int. Cl.⁵: G06F 7/22, G06F 5/06

(22) Date of filing: 21.12.89

(20) Priority: 27.01.89 US 303699

(43) Date of publication of application:
01.08.90 Bulletin 90/31

(84) Designated Contracting States:
DE FR GB

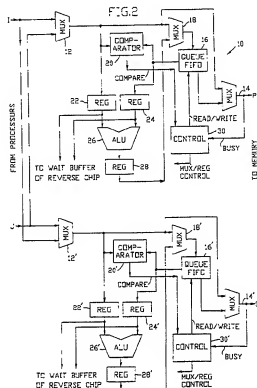
(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: **Hsu, Yarsun**
45 Deerfield Lane South
Pleasantville NY 10570(US)

(74) Representative: **Jost, Ottokarl, Dipl.-Ing.**
Schönaicher Strasse 220
D-7030 Böblingen(DE)

(54) Single-fifo high speed combining switch.

(57) A combining switch 10 includes a two input multiplexer 12 which receives I and J inputs from data processors and directs one of the incoming messages, if there are no contentions or congestions at a switch output port 14 and a Queue FIFO 16 is empty, directly to the output port 14 for transmission to one of a plurality of memory modules. If the output port 14 is busy and the Queue 16 is empty the incoming message is routed to the Queue FIFO 16 for storage. If the Queue FIFO 16 is not empty the incoming message is first compared by a comparator 20 to all existing messages stored in the Queue FIFO 16 to determine if the incoming message is destined for a memory address which already has a queued message. If no match is determined by comparator 20 the incoming message is routed to the Queue FIFO 16 for storage. If comparator 20 determines that the memory address and operation type of the incoming message matches that of a message already stored in the Queue FIFO 16 both the incoming message and the queued message are applied to a message combining ALU 26. The ALU 26 generates a combined message which is stored at the same Queue 16 location as the queued message which generated a comparison match with the incoming message.



SINGLE-FIFO HIGH SPEED COMBINING SWITCH

This invention relates generally to data switching apparatus and, in particular, to a high speed data combining switch which employs, for each half, a single first-in-first-out (FIFO) buffer having an output coupled to a switch output port.

Some multi-processor data processing systems include a number of data processors coupled to a number of memory modules through an interconnection network. The interconnection network may employ an Omega-type switch which includes log-(n) stages of $n/2$ two-by-two switches, where n represents the number of ports being serviced by the switch. One type of switch is known as a combining switch which is used to combine multiple messages which are addressed to the same memory location in order to reduce the number of accesses to that memory location. By combining messages the effects of "hot spot" loading are reduced and the bandwidth of the interconnection network is increased. A decombin switch is subsequently employed to "decombine" responses from memory modules and transmit the responses back to the processors.

Fig. 1 illustrates a conventional 2×2 combining switch 1 comprised of two substantially identical halves. For convenience only one half of the switch will be discussed, the corresponding structure in the other switch half being designated by a primed reference number. Each switch half includes two FIFO register files, one being known as a Chute FIFO 2 and the other being known as a Queue FIFO 3. The Chute and Queue FIFOs each have an equal number of storage locations and are employed to store messages before transmission to the network of memory modules (not shown). Typically, if there are no contentions or congestions at the switch output port 4 and the Queue 3 is empty, incoming processor messages from input ports 1 and 2 are routed directly to the output port 4 via a multiplexer 5. If the Queue 3 is not empty the incoming message is temporarily stored in an input register 6 and compared by a comparator 7 to all existing messages in the Queue to determine if the incoming message is directed to a memory location already associated with a queued message. If a match is not found the incoming registered message is stored in the next available location within the Queue FIFO 3. If a match is detected by the comparator 7 the incoming message is stored instead in the Chute FIFO 2 at a location corresponding to the storage location of the matched message in the Queue 3. Subsequently both the Chute and Queue messages are directed to an arithmetic logic unit (ALU) 8, via ALU input registers 9a and 9b, to combine and generate a single message. In-

formation required for decombin the message on its return from the memory module is sent to a Wait Buffer in an associated decombin switch (not shown).

One significant disadvantage of such conventional combining switches is that the Chute FIFO register file occupies a significant portion of available integrated circuit area. For example, it can be shown that the Chute FIFO 2 can occupy thirty six percent of the data path area as compared to approximately forty five percent for the Queue and ten percent for the ALU. This significant area requirement, and the associated power requirement, for the Chute is especially disadvantageous if the majority of messages sent through the network are not combinable, resulting in only infrequent use of the Chute FIFO.

Another significant disadvantage of such conventional combining switches is that all output from the Queue, whether or not there is a corresponding entry in the Chute, is directed through the ALU. Thus, some finite amount of time is required for the message to pass through the ALU even for those messages which are not combined.

Typically an interconnection network is comprised of a plurality of 2×2 combining switches, such as an 8×8 network. It can therefore be appreciated that an improved packing density, higher speed and reduced power consumption of each of the 2×2 switches would result in an overall improvement in network performance.

It is therefore one object of the invention to provide a combining switch which operates at a higher speed than conventional combining switches.

It is another object of the invention to provide a combining switch which, for each switch half, includes only a Queue FIFO register and which directs messages directly from the Queue FIFO to the switch output port.

It is still another object of the invention to provide a combining switch which has a significant reduction in required integrated circuit surface area, which requires less operating power, and which operates at a higher speed than conventional combining switches.

The foregoing problems are overcome and the objects of the invention are realized by a data switching apparatus, specifically a combining switch having two halves, each of which includes an input port, an output port, a Queue FIFO, a comparator and an ALU. The input port receives data such as messages from data processors and directs incoming messages, if the output port is not busy and the Queue FIFO is empty, directly to the

output port for transmission. If the output port is busy and the Queue FIFO is empty the incoming message is routed to the Queue FIFO for storage. If the Queue FIFO is not empty the incoming message is first compared by the comparator to all existing messages stored in the Queue FIFO to determine if the incoming message is destined for transmission to a memory location which already has a queued message. If no match is determined by the comparator the incoming message is routed to the Queue FIFO for storage. If the comparator determines that the destination location and typically also the operation type of an incoming message matches that of a message already stored in the Queue FIFO both the incoming message and the queued, matching message are applied to the message combining ALU. The ALU generates a combined message which is stored at the same Queue FIFO location as the queued message which generated the comparison match with the incoming message.

In accordance with a method of the invention there is disclosed a method of operating a message combining switch in a data processing system of the type which includes a plurality of data processors which are coupled to a plurality of memory locations through a switching network, the data processors generating messages relative to identified ones of the memory locations. The message combining switch includes two halves each of which has a message storage unit, an input port and an output port. The method includes the steps of receiving a message from the input port and, if the message storage unit has at least one message stored within, comparing an identification of a memory location and an operation type associated with the received message to the identification of memory locations and operation types associated with messages stored in the message storage unit. If the memory location identification and operation type associated with one of the stored messages is determined to be equal to the memory location identification and operation type associated with the received message the method further includes the steps of combining the received message and the stored message to generate a combined message and replacing the stored message with the combined message.

The above set forth and other features of the invention will be made more apparent in the ensuing Detailed Description of the Invention when read in conjunction with the attached Drawing, wherein:

Fig. 1 is a simplified block diagram of a forward path of a 2 X 2 combining switch of the prior art having both Queue and Chute FIFO registers and ALUs through which all output of the Queue FIFOs are directed; and

Fig. 2 is a simplified block diagram of for-

ward path of a 2 X 2 combining switch which, in accordance with the invention, includes for each half only a single FIFO register, specifically the Queue FIFO, which has an output directly coupled to an output port of the switch.

Referring now to Figure 2 there is shown a forward path of a 2 X 2 combining switch 10 constructed in accordance with the invention, it being realized that the switch includes two halves which are constructed in substantially identical fashion. As such, only the upper half of the switch 10 will be discussed, corresponding structure of the lower half of the switch 10 being indicated with a primed reference numeral. Switch 10 includes two input nodes or ports which are coupled to a two input multiplexer 12 which receives I and J message inputs from data processors (not shown) either directly or through a data concentrator. If the switch 10 is located at one of the inner stages of a log(n) switching network the I and J inputs are coupled to the outputs of a combining switch 10 of a previous stage. Multiplexer 12 directs one of the incoming messages, if there are no contentions or congestions at a switch output node 14, and a Queue FIFO 16 is empty, directly to the output port 14 and eventually to one of a plurality of memory modules (not shown). If the output port 14 is busy and the Queue 16 is empty the incoming message is routed to the Queue FIFO 16 for storage via a two input multiplexer 18. However, if the Queue FIFO 16 is not empty, indicating that other outgoing processor messages are stored therein, at least an address portion and more typically both the address and an operation code portion of the incoming message are first compared by a comparator 20 to corresponding portions of all existing messages stored in the Queue FIFO 16. A determination is thus made if the incoming message is destined for a memory address location which already has a queued message. As was stated, in addition to comparing the address location portion or field of the message the comparator 20 typically also compares the operation type portion or field of the message such that only those messages which are directed to the same memory location and which perform the same type of operation, such as READ, WRITE or FETCH_AND_ADD, are combined. So long as the Queue FIFO 16 is not empty this comparison occurs whether or not the output port 14 is busy. If no match is determined by comparator 20 between the memory address and the operation type associated with the received message and the memory addresses and operation types associated with the queued messages the received message is routed through multiplexer 18 to the Queue FIFO 16 for storage. If comparator 20 determines that the memory address and operation type of the incoming message matches that of a

message already stored in the Queue FIFO 16 both the incoming message and the matching queued message are each temporarily stored in an associated register 22 and 24, respectively, for application to a message combining ALU 26. The registered messages are also supplied to a Wait Buffer of an associated message decombining switch (not shown) for later decombination of a message returned from the memory. The ALU 26 generates a combined memory module message which is temporarily stored by ALU output register 28 and which is applied to a second input of multiplexer 18 for storage within the Queue 16. As an example, if both the received message and a queued message indicate a FETCH_AND_ADD operation at the same memory address, the ADD operand of each message are summed by the ALU 26 to generate a single message to that memory location.

In accordance with one aspect of the invention, the combined message from ALU 26 is stored at the same Queue FIFO 16 location as the existing Queue message which generated a comparison match with the incoming message. Thus, the existing message is over-written and replaced by the combined message. Subsequently the queued messages are extracted from the Queue 16 in a first-in/first-out manner for application to the output port 14 as the output port 14 becomes available for transmission to the memory modules or further stages of the switching network.

The switch 10 also includes a control logic block 30 which is responsive to a comparator 20 output signal and a busy condition of the output port 14 to control the operation, in the manner described above, of the FIFO 16 and the various multiplexers and registers. In addition, the switch 10 includes further logic for determining if the incoming message should be directed to port P or Q. The switch typically also includes logic including protocol signals for communicating with preceding and following 2 X 2 switches.

As can readily be seen the combining switch 10 of the invention eliminates both of the Chute FIFOs 2 of the conventional switch of Fig. 1. This elimination of the Chute FIFOs furthermore eliminates, for example, eight transistors per FIFO storage cell. Assuming a six storage location deep by four word wide FIFO, each word being 32 bits in length, a total of 6,144 transistors are eliminated for the one half of the combining switch or a total of 12,288 transistors for entire combining switch 10. As a result, a significant savings in integrated circuit surface area and combining switch power consumption is achieved.

Furthermore, in that incoming messages are routed through the ALU 26 only if there is a match with a queued message a considerable speed advantage is realized over the conventional combin-

ing switch of Fig. 1. That is, the Queue 16 output is coupled directly to the output port 14 multiplexer instead of being coupled to the input of the ALU 26. In the conventional combining switch of Fig. 1 all outgoing messages are sent through the ALU to the output port from the Queue and Chute FIFOs regardless of whether the message requires combination. As such, the conventional combining switch incurs for each output message a propagation delay associated with passage through the ALU.

While the invention has been particularly shown and described with respect to a preferred embodiment thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing from the spirit and scope of the invention.

Claims

1. Digital switching apparatus having an input node means (I, J) and an output node means (P, Q), comprising:
 - storage means (16) having a plurality of storage locations for storing data received from the input node means (I, J) prior to transmission of the data to the output node means (P, Q);
 - comparator means (20) for comparing the received data to all previously received data, if any, which is stored within the storage means (16), the comparator means (20) having an output for indicating if at least one element of the received data matches at least one element of the stored data;
 - combining means (26) having a first input coupled to the received data and a second input coupled to the storage means (16) for combining stored data with the received data, the combining means (26) being responsive to the comparator means (20) output for generating at an output thereof combined data, the combined data being a combination of the received data and the stored data which generated a match with the received data; and
 - means (30) for directing the combined data from the output of the combining means (26) to the storage means (16) for storage at a location wherein the stored data which generated a match with the received data is stored.

2. Digital switching apparatus as set forth in Claim 1 which is included in a data processing system of the type which includes a plurality of data processors which are coupled to a plurality of memory modules through a switching network, the data processors generating messages for storage within a particular one of the memory modules, the input node means (I, J) being coupled to at least one data processor and the output node means (P,

Q) being coupled to at least one memory module.

3. Digital switching apparatus as set forth in Claim 1 or 2 and further comprising first coupling means (MUX) for coupling an output of the storage means (16) to the output node means (P, Q) for providing stored data thereto.

4. Digital switching apparatus as set forth in one of Claims 1 to 3 and further comprising second coupling means (12) for coupling the input node (I, J) means to the output node means (P, Q) if the storage means (16) is empty and if the output node means (P, Q) is available for use.

5. Digital switching apparatus as set forth in one of Claims 1 to 4 and further comprising third coupling means (18) for coupling the input node means (I, J) to the storage means (16) for storing the received data at an available storage location within the storage means (16), the third coupling means (18) being responsive to the comparator means (20) output for coupling the received data from the input node means (I, J) to the storage means (16) when the comparator means (20) output indicates that the at least one element of the received data does not match the at least one element of the stored data.

6. Digital switching apparatus as set forth in one of Claims 1 to 5 wherein the received data is expressive of a message generated by one of a plurality of data processors for storage within one of a plurality of memory modules and wherein the at least one element of data is expressive of an identification of a storage location address of one of the memory modules.

7. Digital switching apparatus as set forth in one of Claims 1 to 6 wherein the storage means (16) comprises a first-in/first-out storage means.

8. Digital switching apparatus as set forth in one of Claims 1 to 7 wherein the combining means (26) comprises an arithmetic/logic unit.

9. In a data processing system of the type which includes a plurality of data processors which are coupled to a plurality of memory locations through a switching network, the data processors generating messages relative to specific ones of the memory locations, a method of operating a message combining switch comprised of a message storage means (16) and an input node means (I, J) coupled to at least one data processor and an output node means (P, Q) coupled to the memory locations, the method comprising the steps of: receiving a message from the input port means (I, J); if the message storage means (16) has at least one message stored within, comparing (20) at least an identification of a memory location and a message operation type associated with the received message to the identification of a memory location and a message operation type associated with messages stored in the mes-

sage storage means (16);

if at least the memory location and operation type associated with one of the stored messages is determined to be equal to the memory location and operation type associated with the received message, combining (26) the received message and the stored message to generate a combined message; and

replacing (30) the stored message with the combined message.

10. A method as set forth in Claim 9 and further comprising a step of, when the memory location and/or the operation type associated with each of the stored messages is determined not to be equal to the memory location and/or the operation type associated with the received message, storing the received message at an available storage location within the message storage means (16).

11. A method as set forth in Claim 9 or 10 wherein the step of receiving includes an additional step of, when the message storage means (16) has no messages stored within and when the output port means is available for use, coupling the received message to the output node means (P, Q) for transmission therefrom.

12. A method as set forth in one of Claims 9 to 11 and further comprising a step of transferring messages from the message storage means (16) to the output node means (P, Q) for transmission therefrom, the step of transferring being accomplished such that the first message stored within the message storage means (16) is the first message transferred out of the message storage means (16).

FIG.1 (PRIOR ART)

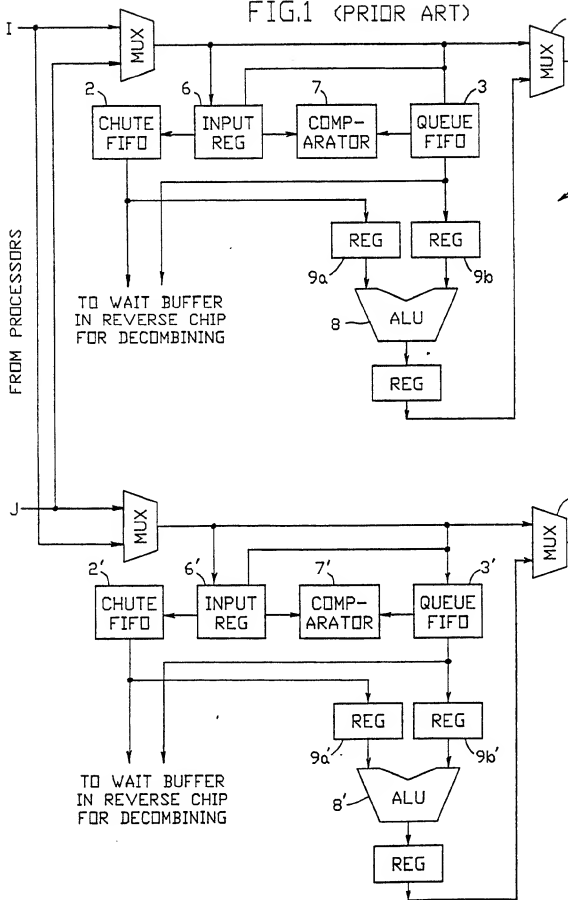


FIG. 2

